

U.S.P.S. Express Mail Label No.: EV 303 831 368 US

Date of Deposit: October 24, 2003

Attorney Docket No. 14920US01

SYSTEM AND METHOD FOR DESIGNING DATA STRUCTURES

CROSS-REFERENCE TO RELATED APPLICATIONS

[01] [Not Applicable]

FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[02] [Not Applicable]

[MICROFICHE/COPYRIGHT REFERENCE]

[03] [Not Applicable]

BACKGROUND OF THE INVENTION

[04] First-in-first-out (FIFO) data structures have been employed in myriad of applications including, for example, very large scale integration (VLSI). Generally, FIFO data structures may be broadly classified as either synchronous or asynchronous.

[05] A synchronous FIFO data structure may be, for example, a FIFO data structure in which a single clock controls the write port and the read port of the FIFO data structure. A typical objective of the synchronous FIFO design is to provide a mechanism for buffering data and subsequently evacuating the buffered data in the order of arrival.

[06] An asynchronous FIFO data structure may be, for example, a FIFO data structure in which a first clock is used to control the write port and a second clock is used to control the read port of the FIFO data structure. The first clock and the second clock may have completely arbitrary phase relationships. In addition to acting as a buffer, the asynchronous FIFO design has been used as a mechanism that reliably transfers data across asynchronous clock domains.

[07] Conventional asynchronous FIFO data structures may be designed for depths of 2^n in which n is an integer. However, being restricted to such depths may be the source of a number of inefficiencies. For example, if an application may be optimized for a FIFO depth of 9, then the use of a FIFO data structure with a depth of 2^4 (i.e., 16) would not fully use all of the available buffer space and would needlessly consume valuable real estate on a printed circuit board or an integrated system (e.g., an integrated chip). The excessive FIFO data structure design may also disrupt stringent fan-out constraints on clock and data signals.

[08] Further limitations and disadvantages of conventional and traditional approaches will become apparent to one of ordinary skill in the art through comparison of such systems with at least some aspects of the present invention as set forth in the remainder of the present application with reference to the drawings.

BRIEF SUMMARY OF THE INVENTION

[09] Aspects of the present invention may be found in, for example, systems and methods for designing data structures. In one embodiment, the present invention may provide an asynchronous first-in-first-out (FIFO) data structure. The data structure may include, for example, a FIFO memory having a depth d in which d is an integer, and a code generator coupled to the FIFO memory. The code generator may provide, for example, a first code sequence of length $2d$. The first code sequence may have a circular property and a Hamming length of one for any two consecutive codes of the first code sequence. The first code sequence may be generated from a second code sequence by removing one or more pairs of mirrored codes of the second code sequence.

[10] In another embodiment, aspects of the present invention may be found in a method for designing an asynchronous data structure. The method may include, for example, one or more of the following: writing data to and reading data from a memory of depth d in which d is not equal to a value 2^n and in which d and n are integers; reducing a first Gray-code sequence of length 2^n into a second Gray-code sequence of length $2d$ by removing one or more pairs of mirrored Gray-codes from the first Gray-code sequence; and using Gray codes of the second Gray-code sequence as Gray-code write pointers or Gray-code read pointers.

[11] In yet another embodiment, aspects of the present invention may provide a method that reduces a Gray-code sequence for use with a data structure of arbitrary depth. The method may include, for example, one or more of the following: listing a plurality of codes in a first Gray-code sequence; deleting one or more first codes from the first Gray-code sequence; deleting one or more second codes from the first Gray-code sequence, each second code having identical bits as a corresponding first code except for a most significant bit; and forming a second Gray-code sequence from remaining codes of the first Gray-code sequence.

[12] In another embodiment, aspects of the present invention may be found in a method for designing an asynchronous data structure. The method may include, for example, one or more of the following: writing data to and reading data from a memory of depth d in which d is not equal to a value 2^n and in which d and n are integers; reducing a first code sequence of

length 2^n into a second code sequence of length $2d$ by removing one or more pairs of mirrored codes from the first code sequence; and using codes of the second code sequence as code write pointers or code read pointers.

[13] In yet another embodiment, aspects of the present invention may provide a method that reduces a code sequence for use with a data structure of arbitrary depth. The method may include, for example, one or more of the following: listing a plurality of codes in a first code sequence; deleting one or more first codes from the first code sequence; deleting one or more second codes from the first code sequence, each second code having identical bits as a corresponding first code except for a most significant bit; and forming a second code sequence from remaining codes of the first code sequence.

[14] These and other features and advantages of the present invention may be appreciated from a review of the following detailed description of the present invention, along with the accompanying figures in which like reference numerals refer to like parts throughout.

BRIEF DESCRIPTION OF THE DRAWINGS

[15] FIG. 1 shows a block diagram illustrating an embodiment of an asynchronous first-in-first-out (FIFO) data structure according to the present invention.

[16] FIG. 2 shows a flow chart illustrating an embodiment of a process that generates a Gray-code sequence according to the present invention.

[17] FIG. 3 shows an embodiment of a Gray-code sequence according to the present invention.

[18] FIG. 4 shows an embodiment of a method for reducing a Gray-code sequence according to the present invention.

[19] FIG. 5 shows another embodiment of a method for reducing a Gray-code sequence according to the present invention.

[20] FIGS. 6A-B show two more embodiments of a method for reducing a Gray-code sequence according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[21] Some aspects of the present invention may be found, for example, in systems and methods for designing data structures such as, for example, first-in-first-out (FIFO) data structures. Some embodiments according to the present invention may be used with asynchronous FIFO data structures or synchronous FIFO data structures. The depth of the FIFO data structures may be arbitrary and need not be limited to depths of 2^n in which n is an integer.

[22] Some embodiments according to the present invention may employ, for example, a Gray-code sequence that has been adapted for use with FIFO data structures of any arbitrary depth. A mirroring property of Gray code sequences may be used to effectively generate codes that are less than 2^n in length while still possessing properties suitable for use with FIFO design (e.g., asynchronous FIFO design).

[23] Some embodiments according to the present invention may employ, for example, a code sequence that has been adapted for use with FIFO data structures of any arbitrary depth. The code sequence may be characterized by one or more properties including, for example, a closed property and a Hamming distance of one.

[24] FIG. 1 shows a block diagram illustrating an embodiment of an asynchronous FIFO data structure according to the present invention. The asynchronous FIFO data structure 100 may generally be grouped into a write clock domain 110 and a read clock domain 120. The write clock domain 110 may include, for example, a write clock. The read clock domain 120 may include, for example, a read clock. The write clock and the read clock need not have a phase relationship or a time relationship (e.g., a period relationship). The clocks may be asynchronous. In addition, one or both of the clocks may not be free running.

[25] The write clock domain 110 may include, for example, a Gray-code generator 130, a Gray-code write pointer 140, a Gray-to-binary converter 150, write clock logic 170 related to generating an empty, full or almost full signal, and synchronizers 180. The read clock domain 120 may include, for example, a Gray-code generator 190, a Gray-code read pointer

200, a Gray-to-binary converter 210, read clock logic 220 related to generating an empty, full or almost full signal and synchronizers 230.

[26] The asynchronous FIFO data structure 100 may also include, for example, a FIFO RAM 160 that is part of the write clock domain 110 and the read clock domain 120. The FIFO RAM 160 may include inputs such as, for example, a binary write pointer input, a binary read pointer, a write data port and a write clock. The FIFO RAM 160 may include an output such as, for example, a read data port. The FIFO RAM 160 may have a FIFO depth of any arbitrary depth d in which d is an integer and need not be limited merely to depths of 2^n in which n is an integer. In some embodiments, whereas the FIFO depth is d or $2d$, the Gray-code sequence length is $2d$.

[27] In operation, according to the write clock in the write clock domain 110, a Gray-code write pointer 140 may point to the next Gray-code address to be written. The Gray-to-binary converter 150 may convert the Gray code to a binary code $WrPtr$ which points to a particular binary address in the FIFO RAM 160. Write data $WRDATA$ may then be written into the FIFO RAM 160 at the memory location associated with the binary address $WrPtr$. After the data is written into the FIFO RAM 160, the Gray count may be incremented according to the Gray-code generator 130. The logic 170 may generate an empty signal, a full signal or an almost full signal depending upon, for example, a relative state of the read pointer and the write pointer. The synchronizers 180 may be used, for example, to synchronize the Gray-code read pointer into the write clock domain 110.

[28] According to the read clock in the read clock domain 120, a Gray-code read pointer 200 may point to the current Gray-code address to be read. The Gray-to-binary converter 210 may convert the Gray code to a binary code $RdPtr$ which points to a particular binary address in the FIFO RAM 160. Read data $RDDATA$ may then be read out of the FIFO RAM 160 from the memory location associated with the binary address $RdPtr$. After the data is read out of the FIFO RAM 160, the Gray count may be incremented according to the Gray-code generator 190. The logic 220 may generate an empty signal, a full signal or an almost full signal depending upon, for example, a relative state of the read pointer and the write

pointer. The synchronizers 230 may be used to synchronize the Gray-code write pointer into the read clock domain 120.

[29] In one embodiment, the asynchronous FIFO data structure 100 may provide resilience towards meta-stability situations that may arise, for example, due to the reception of a read/write pointer from a different clock domain. Gray codes transmitted via synchronizers may provide the resilience. The synchronization scheme may guarantee that the meta-stability resolution time (e.g., the time taken by the synchronizer register to come to a valid logic value of 0 or 1 from a meta-stable state) may be less than the period of the receiving clock. Thus, there may be a change of only one bit of the pointer that is sent across to the other domain. If the changing value is not captured by the current clock edge due to set-up or hold violations, but meta-stability is resolved by the synchronizers, then the changing value may be captured by the next clock edge of the read clock.

[30] In one embodiment, to design an asynchronous FIFO data structure 100 with a depth d in which d is an integer and in which d does not have a value of 2^n and in which n is an integer, a code counter may generate a code having circular properties and having a Hamming distance equal to one for any two consecutive codes. One example of such a code is the Gray code. In a Gray-code sequence, any two consecutive codes differ only in the change of one bit position. Furthermore, the Gray-code sequence is closed (i.e., the first code and the last code in the sequence also differ only in the change of one bit position). According to some embodiments of the present invention, an asynchronous FIFO data structure 100 with a depth d is supported by a Gray-code sequence of length $2d$.

[31] FIG. 2 shows a flow chart illustrating an embodiment of a process that generates a Gray-code sequence of any length $2d$ according to the present invention. In step 240, a Gray-code sequence is generated. The Gray-code sequence may be have a length (e.g., 2^n) greater than the desired length $2d$. In step 250, the Gray-code sequence is reduced, for example, by removing one or more codes and the corresponding mirrored codes. The removal of pairs of codes (i.e., a code and its corresponding mirrored code) may reduce the Gray-code sequence to the desired length $2d$. In step 260, the reduced code may then be used in the asynchronous FIFO data structure 100. For example, the reduced code of length $2d$ may be used by the

Gray-code generators 130, 190 in generating the Gray-code write pointer and the Gray-code Read pointer.

[32] FIG. 3 shows an embodiment of a Gray-code sequence according to the present invention. The Gray-code sequence may have a sequence length of 16 and each code may include 4-bits. For illustrative purposes, a centerline may be drawn to evenly divide the sequence into a top portion and a bottom portion. The three least significant bits (LSBs) of each code of the top portion may be mirrored in the bottom portion with respect to the centerline. For example, the three LSBs of Gray-code 0100 may be mirrored as 1100 with respect to the centerline, the only difference being the most significant bit (MSB) of each code in the mirrored pair. In another example, the three LSBs of Gray-code 0111 may be mirrored as 1111 with respect to the centerline.

[33] FIG. 4 shows an embodiment of a method for reducing a Gray-code sequence according to the present invention. The Gray-code sequence may be reduced by removing a middle portion centered on the centerline. For example, the middle portion may include three mirrored pairs: 0111 and 1111; 0101 and 1101; and 0100 and 1100. With the removal of the middle portion, the original Gray-code sequence may be reduced from a sequence length of 16 to a sequence length of 10. Even though the sequence has been reduced in length, the sequence may still retain its circular properties and have a Hamming distance equal to one for any two consecutive codes.

[34] FIG. 5 shows another embodiment of a method for reducing a Gray-code sequence according to the present invention. The Gray-code sequence may be reduced by removing a top portion and a corresponding bottom portion (i.e., a mirrored portion with respect to the center line). For example, the top portion and the bottom portion may include three mirrored pairs: 0000 and 1000; 0001 and 1001; and 0011 and 1011. With the removal of the middle portion, the original Gray-code sequence may be reduced from a sequence length of 16 to a sequence length of 10.

[35] FIGS. 6A-B show two more embodiments of a method for reducing a Gray-code sequence according to the present invention. The Gray-code sequence may be reduced by

removing a middle portion and a top portion with a corresponding bottom portion. In FIG. 6A, the middle portion includes a single pair of mirrored codes and the top/bottom portion includes two pairs of mirrored codes. In FIG. 6B, the middle portion includes two pairs of mirrored codes and the top/bottom portion includes a single pair of mirrored codes.

[36] FIGS. 4-6 illustrate, for example, that a particular Gray-code sequence may be reduced a number of different ways to create a number of different reduced sequences. From a plurality of different reduced sequences, an optimal code sequence may be chosen which optimizes the mapping of the transformation function and design of the counters that are used for the read and write pointers.

[37] Some embodiments according to the present invention may have one or more of the following advantages. By releasing the restriction that a FIFO data structure be of depth 2^n , many inefficiencies may be avoided such as, for example, underused FIFO data structure depth. Furthermore, many of the efficiencies such as excessive FIFO data structure depth may become quite substantial as FIFO data structure width or the number of FIFO data structures increases. In addition, by designing a FIFO data structure of any depth, many of the fan-out constraints on clock and data signals may be maintained.

[38] While the present invention has been described with reference to certain embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted without departing from the scope of the present invention. In addition, many modifications may be made to adapt a particular situation or material to the teachings of the present invention without departing from its scope. Therefore, it is intended that the present invention not be limited to the particular embodiments disclosed, but that the present invention will include all embodiments falling within the scope of the appended claims.